

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-328880

(43) 公開日 平成8年(1996)12月13日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46 11/34	3 4 0	7313-5B 7313-5B	G 0 6 F 9/46 11/34	3 4 0 D A H

審査請求 未請求 請求項の数14 O L (全 25 頁)

(21) 出願番号 特願平7-133920

(22) 出願日 平成7年(1995)5月31日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 松本 肇

兵庫県伊丹市瑞原4丁目1番地 三菱電機

株式会社北伊丹製作所内

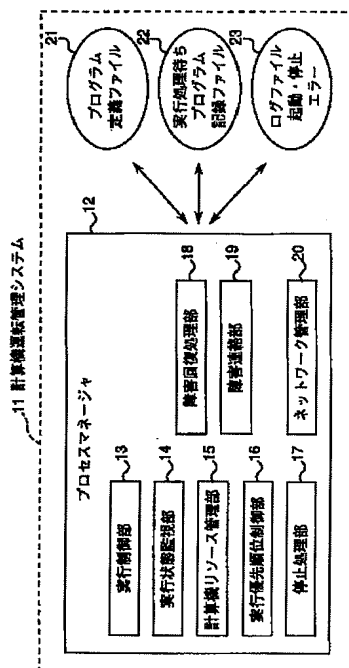
(74) 代理人 弁理士 高田 守 (外4名)

(54) 【発明の名称】 複数のアプリケーションプログラムを同時に実行できるオペレーティングシステムにおける計算機運転管理システム

(57) 【要約】

【目的】 本発明は、同時に複数のプログラムを実行できるオペレーティングシステムにおいて、同時に実行できるプログラムの数を制限、またプログラムの実行状況を監視することで、アプリケーションプログラムを信頼性高く処理し、計算機システムの安定的な運転管理機能を提供するものである。

【構成】 計算機運転管理システムにおいて、同時実行可能なプログラム数を制限する手段と、実行状態を監視する手段と、優先順位を制御する手段と、停止後の処理を制御する手段と、計算機のリソースを管理する手段と、障害時処理を行う手段と、障害通知を行う手段と、ネットワークの管理を行う手段とから構成され、実行プログラムの各種制御情報を記録するプログラム定義ファイルと、実行待ちプログラムの情報を記録する実行処理待ちプログラム記録ファイルと、プログラムのログ情報を記録するログファイルを備える。



【特許請求の範囲】

【請求項1】 計算機の運転管理システムにおいて、複数のアプリケーションプログラムを同時に実行できるオペレーティングシステムに対し、同時に実行できるアプリケーションプログラムの実行数を制限し、監視することによりアプリケーションソフトウェアを信頼性高く安定に運用するための計算機運転管理システム。

【請求項2】 請求項1に記載の計算機の運転管理システムにおいて、上記オペレーティングシステムは、同時に複数のアプリケーションプログラムを実行可能であることと、1つの実行中プログラムから、新たに直接通信または直接管理できる関係のプログラムを生成可能であることと、プログラム起動時に、オペレーティングシステムに対し特に指定しなければ、実行プログラムは、同じ優先順位で平等に扱われ、また実行中のプログラムに対して、優先順位を動的に変更できることと、有限な大きさの仮想記憶領域を持ち、該仮想記憶領域の限界を超えて、プログラムを実行しようとした時は、即座に異常終了となることを特徴とするオペレーティングシステムであり、上記計算機システムは、該オペレーティングシステムとアプリケーションプログラムの間に介在し、上記オペレーティングシステムとアプリケーションプログラム双方に対してインターフェースを持ち、上記オペレーティングシステムに対し、1つのアプリケーションプログラムであるプログラムと、ユーザによりあらかじめ定義される、プログラム種別ごとのプログラムの同時実行可能制限数、プログラムの優先順位、障害回復手順、ネットワーク関連プログラム、停止後処置の情報を記録する手段であるプログラム定義ファイルと、実行処理待ちプログラムの情報を記録する手段である実行処理待ちプログラム記録ファイルと、プログラムの起動、停止、障害時に、ログ情報を記録する手段であるログファイルから構成されることを特徴とする計算機運転管理システム。

【請求項3】 請求項1または請求項2に記載の計算機運転管理システムにおいて、1つのオペレーティングシステム上で実行される1つのアプリケーションプログラム運転管理用プログラムと、管理すべきアプリケーションプログラムを該運転管理用プログラムと直接通信できる関係または直接管理できる関係で実行することにより、アプリケーションプログラムの実行状況を管理、制御できることを特徴とする計算機運転管理システム。

【請求項4】 請求項1または請求項2に記載の計算機運転管理システムにおいて、アプリケーションプログラムの制御方法として、アプリケーションプログラム起動時において、ユーザにより定義されるプログラム種別や計算機ごとの同時に実行可能なプログラム数の制限数を記録したプログラム定義ファイルと、複数プログラムの同時実行時において、上記プログラム定義ファイルから読み出した同時実行可能プログラム数の制限数により、

プログラム種別や計算機ごとにプログラムの実行数を上記オペレーティングシステムに対し制限することにより、計算機のリソースを有効に利用し、アプリケーションプログラムの安定した運用を実現する実行制御部と、同時に実行できなかったプログラムの情報を一定時間後に再起動するために、一時的に記録する処理待ちプログラム記録手段と、プログラム起動後、起動したプログラムの名前、日時、計算機名を含む起動ログを記録するログファイルとを有することを特徴とする計算機運転管理システム。

【請求項5】 請求項1または請求項2に記載の計算機運転管理システムにおいて、アプリケーションプログラムの状態監視方法として、定期的にアプリケーションプログラムへメッセージの送受信を行なう手段と、アプリケーションプログラムからの返信メッセージを受信し、受信内容を確認することにより、該アプリケーションプログラムが実行状態であることを確認する手段と、該アプリケーションからメッセージの返信がない時はプログラムに異常が生じたと判断し、所定の障害回復手順を行う障害回復処理部と所定の障害時の連絡を行う障害連絡部に連絡する手段とを備える実行状態監視部を有することを特徴とする計算機運転管理システム。

【請求項6】 請求項5に記載の計算機運転管理システムにおいて、アプリケーションプログラムの異常発生時の障害回復方法として、ユーザによりあらかじめ定義される異常終了した場合の障害対応方法を記録するプログラム定義ファイルと、障害発生時に上記プログラム定義ファイルに記録された障害対応方法に基づき障害回復を実行する回復処理実行手段を備える障害回復処理部を有することを特徴とする計算機運転管理システム。

【請求項7】 請求項5に記載の計算機運転管理システムにおいて、計算機上で発生した障害をオペレータに連絡する方法として、ネットワークで結合した全ての計算機にある運転管理システムの管理下のアプリケーションソフトウェアのログファイルを集める手段と、ユーザにより定義される障害連絡方法の情報を記録したプログラム定義ファイルと、データをオペレータに解かりやすい形式に加工し、表示する手段、自動的にアラーム発報、電子メール、コンソール、プリンタ、ファイルにより連絡する手段とを備える障害連絡部を有することを特徴とする計算機の運転管理システム。

【請求項8】 請求項4または請求項5に記載の計算機運転管理システムにおいて、上記オペレーティングシステムにおいて、複数のプログラムを同時に実行するための方法として、実行制御、プロセス起動のステップが終了後、実行状態監視ステップにて、新たなプログラムの起動要求を受けつけ、後から実行したプログラムを前に実行したプログラムとは独立して平行させて起動することにより、同時に複数のプログラムを実行することを特徴とする計算機運転管理システム。

【請求項9】 請求項1または請求項2または請求項3に記載の計算機運転管理システムにおいて、アプリケーションプログラムの実行優先順位を制御する方法として、ユーザにより定義されるプログラム種別ごとの実行優先順位を記録するプログラム定義ファイルと、該プログラム定義ファイルから読み出した優先順位に基づいて、上記実行処理待プログラム記録ファイルの中で優先順位に従い、待機プログラムの並べ替えを行い、また実行中のプログラムの実行優先順位と、処理待ちプログラムの実行優先順位を動的に変更し、変更後の優先順位を上記オペレーティングシステムに対し指示することにより、実行中のプログラムおよび待機中のプログラムに対し、優先順位を動的に制御する手段とを備える実行優先順位制御部を有することを特徴とする計算機運転管理システム。

【請求項10】 請求項1または請求項2に記載の計算機運転管理システムにおいて、計算機上で実行中のアプリケーションプログラムの停止処理方法として、オペレーティングシステムからの終了メッセージにより、プログラムが終了したことを認識する手段と、プログラムの終了した時の終了状態を識別する手段と、ユーザにより定義される終了状態に応じて連続して行なうべき後処置の定義を記録するプログラム定義ファイルと、該後処置の定義に従い、後処置を実行する後処置実行手段と、プログラム停止時に、プログラム名、計算機名、停止日時を含む停止ログを記録するログファイルとを備える停止処理部を有することを特徴とする計算機運転管理システム。

【請求項11】 請求項1または請求項2に記載の計算機運転管理システムにおいて、計算機上で実行中のアプリケーションプログラムのリソースを管理する方法として、実行中のアプリケーションプログラムごとに使用できる主記憶装置（メモリ）、補助記憶装置（ディスク）の容量や、演算処理時間の累積値の制限値を記録するプログラム定義ファイルと、定義した主記憶、補助記憶の容量や演算処理時間の制限値を越えてプログラムが実行されたことを検出する手段と、計算機のリソースが制限値を越えて使用された時に障害と判断し、上記障害回復処理部、および上記障害連絡部に連絡する機能を有する手段を備える計算機リソース管理部を有することを特徴とする運転管理システム。

【請求項12】 請求項1または請求項2に記載の計算機運転管理システムにおいて、複数の計算機をネットワークで結合し、計算機間で通信を行ないながら実行するアプリケーションソフトウェアの動作を管理するための方法として、異なる計算機にある運転管理システム間で通信連絡する手段と、ユーザにより定義されるネットワーク上の関連プログラムの情報を記録するプログラム定義ファイルと、該プログラム定義ファイルのネットワーク関連プログラムの情報に従い、ネットワークを利用す

るアプリケーションソフトウェアの動作を管理する手段と、ネットワーク上で関連のあるプログラムの組み合わせを認識する手段とを備える計算機ネットワーク管理部を有することを特徴とする計算機運転管理システム。

【請求項13】 請求項1または請求項2に記載の計算機運転管理システムにおいて、実行制御部、実行状態監視部、計算機リソース管理部、停止処理部は一連の主プログラムの中で連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出されて実行されることを特徴とする計算機運転管理システム。

【請求項14】 請求項1または請求項2に記載の計算機運転管理システムにおいて、実行制御部、実行状態監視部、停止処理部は一連の主プログラムの連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、計算機リソース管理部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出され実行されることを特徴とする計算機運転管理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、1つの計算機上のあるいは、ネットワーク上の複数の計算機上でのアプリケーションプログラムを信頼性高く、高速で処理するためのシステムで、計算機の安定的な運転管理システムに関するものである。

【0002】

【従来の技術】従来の計算機システムは、メモリ、ディスク、CPU、ネットワーク機器等のハードウェアとこれらのハードウェアを効果的に使用するためのオペレーティングシステムソフトウェアと、業務アプリケーションごとに計算機処理手順を記述したプログラムから構成されている。計算機を使用する人は、アプリケーションプログラムをプログラミング言語を用いて記述し、オペレーティングシステムソフトウェアにプログラムの実行を指示する。オペレーティングシステムは、その計算機の持っているメモリ（主記憶装置）、ディスク（補助記憶装置）、CPU（主制御装置）、ネットワーク（通信）機器などを効果的に使用して、依頼されたプログラムを1つのみ、または同時に複数個実行する。オペレーティングシステムとしては、汎用計算機で使用されるMVS、ミニコンで使用されるVAX、ワークステーションで使用されるUNIX、パソコンで使用されるMS-DOS、Windowsが世界的に有名で、事実上の標準となっている。

【0003】同時に実行可能な計算機のオペレーティングシステムとして、最近、よく利用されるようになったものにUNIXというオペレーティングシステムがある。

【0004】UNIXは、米国AT&Tにて開発されたオペレーティングシステムであり、ワークステーションを中心とした計算機に広く使用されている。UNIXは、もともとソフトウェアの開発、CAD(Computer Aided Design)等で使用された技術者向けの計算機基本ソフトウェアであり、他のオペレーティングシステムと比較して下記の特徴を持つ。

(1) 同時に複数のプログラムを並列に実行できる。

(2) 1つの実行中プログラムから新しいプログラムを生成することができる。OSによる制限を受けずに同時実行プログラム数が増加する。

(3) 同時に実行しているプログラムは、特に指定がなければ、同じ優先順位で実行され、平等に扱われる。実行中の優先順位を動的に変更することができる。

(4) 仮想記憶領域は、有限の大きさを持つが、この限界を越えてプログラムを実行しようとした時、即座にエラー終了となる。

【0005】

【発明が解決しようとする課題】UNIXでは、1つの計算機で同時に実行できるアプリケーションプログラムの数を制限することができない。従って、同時に実行できるプログラムの数が予想値より多くなった場合、計算機の持っている主記憶装置(メモリ)の容量が不足する。メモリ不足になると、UNIXは、補助記憶装置(ディスク)上に仮想記憶領域を確保することで、メモリ上の記憶内容をディスクに書き込む。ディスク上に移動した場合、そのプログラムの実行速度は数十分の一から数百分の一に低下する。実行速度が遅くなれば、リアルタイム性が要求されるシステムではトラブルの原因となる。

【0006】更に、メモリが不足し、補助記憶装置(ディスク)上の仮想記憶領域までも不足した場合、UNIXは、実行中のプログラムの1つを強制終了する。強制終了すべきプログラムの優先順位を指定することができないため、信頼性が要求されるプログラムも、信頼性が要求されないプログラムも同じ確率で強制終了する。プログラムが強制終了するとシステムの機能の一部が停止する。

【0007】UNIXでは、プログラムの起動時、正常終了時、異常終了時に、プログラムがどのような状態変化をしたか、その時間がいつか、出力エラーメッセージは何かを集中管理的に記録する手段がない。従って、システムにトラブルが発生した場合に、原因となるプログラムを特定するための手掛かりがない。また、一部には出力されるメッセージもあるが、マシン別、出力機器、出力方法等も統一されていないため、トラブル解析作業が手間取る。

【0008】UNIXでは、実行された複数のプログラムは平等に扱われ、割当てられる実行時間(CPUタイム)の量をコントロールできない。したがって、CPU

を異常消費するプログラムや、メモリを異常消費するプログラムにより、レスポンスタイムの悪化、システム全体のダウンを招くことがある。

【0009】以上に示したような欠点は、UNIXを用いた計算機を複数ネットワークにより結合した大規模システムでは、多発しやすく、その原因追求も複雑になる。また、これらの不具合そのものが発生したことを、オペレータに即時に連絡するための手段もなく、回復作業が遅れる。

【0010】本発明は上記のような課題を持つオペレーティングシステム(UNIXなど)上で動作する計算機運転管理システムを提供する。

【0011】

【課題を解決するための手段】本発明は、複数のアプリケーションプログラムを同時に実行できるオペレーティングシステムに対し、アプリケーションの実行数を制限し、またアプリケーションプログラムの実行、計算機リソース、ネットワークを監視することにより、アプリケーションソフトウェアを信頼性高く安定に運用できる計算機運転管理システムを提供するものである。

【0012】本発明の特許請求の範囲の請求項2に記載の計算機の運転管理システムにおいて、オペレーティングシステムとアプリケーションの間に介在し、オペレーティングシステムとアプリケーションプログラム双方に対してインターフェースを持ち、オペレーティングシステムに対し、1つのアプリケーションプログラムであるプログラムと、ユーザによりあらかじめ定義される、プログラム種別ごとのプログラムの同時実行可能制限数、プログラムの優先順位、障害回復手順、ネットワーク関連プログラム、停止後処置の情報を記録する手段であるプログラム定義ファイルと、実行処理待ちプログラムの情報を記録する手段である実行処理待ちプログラム記録ファイルと、プログラムの起動、停止、障害時に、ログ情報を記録する手段であるログファイルから構成されることを特徴とする。

【0013】本発明の特許請求の範囲の請求項3に記載の計算機運転管理システムにおいて、1つのオペレーティングシステム上で実行される1つのアプリケーションプログラム運転管理用プログラムと、管理すべきアプリケーションプログラムを該運転管理用プログラムと直接通信できる関係または直接管理できる関係で実行することにより、アプリケーションプログラムの実行状況を管理、制御できることを特徴とする。

【0014】本発明の特許請求の範囲の請求項4に記載の計算機運転管理システムにおいて、アプリケーションプログラムの制御方法として、アプリケーションプログラム起動時において、ユーザにより定義されるプログラム種別や計算機ごとの同時に実行可能なプログラム数の制限数を記録する手段であるプログラム定義ファイルと、複数プログラムの同時実行時において、上記プログ

ラム定義ファイルから読み出した同時実行可能プログラム数の制限数により、プログラム種別や計算機ごとにプログラムの実行数を上記オペレーティングシステムに対し制限することにより、計算機のリソースを有効に利用し、アプリケーションプログラムの安定した運用を実現する実行制御部と、同時に実行できなかったプログラムの情報を一定時間後に再起動するために、一時的に記録する手段である処理待ちプログラム記録ファイルと、プログラム起動後、起動したプログラムの名前、日時、計算機名を含む起動ログを記録する手段であるログファイルとを有する。

【0015】本発明の特許請求の範囲の請求項5に記載の計算機運転管理システムにおいて、アプリケーションプログラムの状態監視方法として、定期的にアプリケーションプログラムへメッセージの送受信を行なう手段と、アプリケーションプログラムからの返信メッセージを受信し、受信内容を確認することにより、該アプリケーションプログラムが実行状態であることを確認する手段と、該アプリケーションからメッセージの返信がない時はプログラムに異常が生じたと判断し、所定の障害回復手順を行う障害回復処理部と所定の障害時の連絡を行う障害連絡部に連絡する手段とを備える実行状態監視部を有する。

【0016】本発明の特許請求の範囲の請求項6に記載の計算機運転管理システムにおいて、アプリケーションプログラムの異常発生時の障害回復方法として、ユーザによりあらかじめ定義される異常終了した場合の障害対応方法を記録する手段であるプログラム定義ファイルと、障害発生時に記録された障害対応方法に基づき障害回復を実行する回復処理実行手段を備える障害回復処理部を有する。

【0017】本発明の特許請求の範囲の請求項7に記載の計算機運転管理システムにおいて、計算機上で発生した障害をオペレータに連絡する方法として、ネットワークで結合した全ての計算機にある運転管理システムの管理下のアプリケーションソフトウェアのログファイルを集める手段と、ユーザによりあらかじめ定義される障害時の連絡手段を記録する手段であるプログラム定義ファイルと、データをオペレータに解かりやすい形式に加工し、表示する手段、自動的にアラーム発報、電子メール、コンソール、プリンタ、ファイルにより連絡する手段とを備える障害連絡部を有する。

【0018】本発明の特許請求の範囲の請求項8に記載の計算機運転管理システムにおいて、上記オペレーティングシステムにおいて、複数のプログラムを同時に実行するための方法として、実行制御、プロセス起動のステップが終了後、実行状態監視ステップにて、新たなプログラムの起動要求を受けつけ、後から実行したプログラムを前に実行したプログラムとは独立して平行させて起動することにより複数のプログラムを同時に実行できる

ことを特徴とする。

【0019】本発明の特許請求の範囲の請求項9に記載の計算機運転管理システムにおいて、アプリケーションプログラムの実行優先順位を制御する方法として、ユーザにより定義されるプログラム種別ごとの実行優先順位を記録するプログラム定義ファイルと、該プログラム定義ファイルから読み出した優先順位に基づいて、上記処理待ちプログラム記録ファイルの中で優先順位に従い、待機プログラムの並べ替えを行い、また実行中のプログラムの実行優先順位と、処理待ちプログラムの実行優先順位を動的に変更し、変更後の優先順位を上記オペレーティングシステムに対し指示することにより、実行中のプログラムおよび待機中のプログラムに対し、優先順位を動的に制御する手段とを備える実行優先順位制御部を有する。

【0020】本発明の特許請求の範囲の請求項10に記載の計算機運転管理システムにおいて、計算機上で実行中のアプリケーションプログラムの停止処理方法として、オペレーティングシステムからの終了メッセージにより、プログラムが終了したことを認識する手段と、プログラムの終了した時の終了状態を識別する手段と、ユーザによりあらかじめ定義される終了状態に応じて連続して行なうべき後処置の定義を記録するプログラム定義ファイルと、該プログラム定義ファイルの定義に従い、後処置を実行する後処置実行手段と、プログラム停止時に、プログラム名、計算機名、停止日時を含む停止ログを記録する手段であるログファイルとを備える停止処理部を有する。

【0021】本発明の特許請求の範囲の請求項11に記載の計算機運転管理システムにおいて、計算機上で実行中のアプリケーションプログラムのリソースを管理する方法として、実行中のアプリケーションプログラムごとに使用できる主記憶装置（メモリ）、補助記憶装置（ディスク）の容量や、演算処理時間の累積値の制限値を記録する手段であるプログラム定義ファイルと、定義した主記憶、補助記憶の容量や演算処理時間の制限値を越えてプログラムが実行されたことを検出する手段と、計算機のリソースが制限値を越えて使用された時に障害と判断し、上記障害回復処理部、および上記障害連絡部に連絡する機能を有する手段とを備える計算機リソース管理部を有する。

【0022】本発明の特許請求の範囲の請求項12に記載の計算機運転管理システムにおいて、複数の計算機をネットワークで結合し、計算機間で通信を行ないながら実行するアプリケーションソフトウェアの動作を管理するための方法として、異なる計算機にある運転管理システム間で通信連絡する手段と、ネットワーク上の関連したプログラムの情報を記録する手段であるプログラム定義ファイルと、ネットワークを利用するアプリケーションソフトウェアの動作を管理する手段と、ネットワーク

上で関連のあるプログラムの組み合わせを認識する手段とを備える計算機ネットワーク管理部を有する。

【0023】本発明の特許請求の範囲の請求項13に記載の計算機運転管理システムにおいて、実行制御部、実行状態監視部、計算機リソース管理部、停止処理部は一連の主プログラムの中で連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出されて実行されることを特徴とする。

【0024】本発明の特許請求の範囲の請求項14に記載の計算機運転管理システムにおいて、実行制御部、実行状態監視部、停止処理部は一連の主プログラムの連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、計算機リソース管理部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出され実行されることを特徴とする。

【0025】

【作用】特許請求の範囲の請求項1に記載の計算機の運転管理システムにおいて、複数のアプリケーションプログラムを同時に実行できるオペレーティングシステムに対し、アプリケーションプログラムの実行数を制限し、またアプリケーションプログラムの実行、計算機リソース、ネットワークを監視することにより、アプリケーションソフトウェアを信頼性高く安定に運用できる。

【0026】特許請求の範囲の請求項2に記載の計算機の運転管理システムにおいて、該計算機運転管理システムは、複数のアプリケーションプログラムを同時に実行可能であり、1つの実行中プログラムから、新たに直接通信または直接管理できる関係のプログラムを生成可能であり、プログラム起動時に、オペレーティングシステムに対し特に指定しなければ、実行プログラムは、同じ優先順位で平等に扱われ、また実行中のプログラムに対して優先順位を動的に変更でき、また有限な大きさの仮想記憶領域を持ち、該仮想記憶領域の限界を超えて、プログラムを実行しようとした時は、即座に異常終了となることを特徴とするオペレーティングシステムと、アプリケーションの間に介在し、上記オペレーティングシステムとアプリケーションプログラム双方に対してインターフェースを持ち、上記オペレーティングシステムに対し、1つのアプリケーションプログラムとして動作する。また、プログラム定義ファイルは、ユーザによりあらかじめ定義されるプログラム種別ごとのプログラムの同時実行可能制限数、プログラムの優先順位、障害回復手順、ネットワーク関連プログラム、停止後処置の情報を記録する。実行処理待ちプログラム記録ファイルは、実行の処理待ちプログラムの情報を記録する。ログファイルは、プログラムの起動、停止、障害時に、ログ情報を記録する。

【0027】特許請求の範囲の請求項3に記載の計算機運転管理システムにおいて、1つのオペレーティングシステム上で実行される1つのアプリケーションプログラム運転管理用プログラムと、管理すべきアプリケーションプログラムを該運転管理用プログラムと直接通信できる関係または直接管理できる関係で実行することにより、アプリケーションプログラムの実行状況を管理、制御できることを特徴とする計算機運転管理システム。

【0028】特許請求の範囲の請求項4に記載の計算機運転管理システムにおいて、アプリケーションプログラム起動時において、プログラム定義ファイルは、ユーザにより定義されるプログラム種別や計算機ごとの同時に実行可能なプログラム数の制限数を記録する。複数プログラムの同時実行時において、実行制御部は、上記プログラム定義記録手段から読み出した同時実行可能プログラム数の制限数により、プログラム種別や計算機ごとにプログラムの実行数を上記オペレーティングシステムに対し制限する。これにより、計算機のリソースを有効に利用し、アプリケーションプログラムの安定した運用を実現する。また該実行制御部は、同時に実行できなかったプログラムの情報を一定時間後に再起動するために、一時的に処理待ちプログラムファイルに待機プログラムの情報を記録する。プログラム起動後、該実行制御部は、起動したプログラムの名前、日時、計算機名を含む起動ログをログファイルに記録する。

【0029】特許請求の範囲の請求項5に記載の計算機運転管理システムにおいて、実行状態監視部は、定期的にアプリケーションプログラムへメッセージの送受信を行ない、アプリケーションプログラムからの返信メッセージを受信し、受信内容を確認することにより、該アプリケーションプログラムが実行状態であることを確認する。また、アプリケーションからメッセージの返信がない時はプログラムに異常が生じたと判断し、所定の障害回復手順を行う障害回復処理部と所定の障害時の連絡を行う障害連絡部に連絡する。

【0030】特許請求の範囲の請求項6に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、ユーザによりあらかじめ定義される異常終了した場合の障害対応方法を記録する。障害回復処理部は、障害発生時に、上記障害時処理記録手段から読みだした処理手順に従い、障害回復処理を実行する。

【0031】特許請求の範囲の請求項7に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、ユーザによりあらかじめ定義された連絡方法を記録する。障害連絡部は、ネットワークで結合した全ての計算機にある運転管理システムの管理下のアプリケーションソフトウェアのログファイルを集める。また、上記プログラム定義ファイルから読みだした方法に従い、データをオペレータに解かりやすい形式に加工し、表示し、自動的にアラーム発報、電子メール、コンソール、プリ

ンタ、ファイルにより連絡する。

【0032】特許請求の範囲の請求項8に記載の計算機運転管理システムにおいて、実行制御、プログラム起動のステップが終了後、実行状態監視ステップにて、新たなプログラムの起動要求を受けつけ、後から実行したプログラムを前に実行したプログラムとは独立して平行に起動することにより、複数のプログラムを同時に実行できる。

【0033】特許請求の範囲の請求項9に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、ユーザにより定義されるプログラム種別ごとの実行優先順位を記録する。実行優先順位制御部は、上記プログラム定義ファイルから読み出した優先順位に基づいて、上記処理待プログラム記録ファイルの中で優先順位に従い、待機プログラムの並べ替えを行い、また実行中のプログラムの実行優先順位と、処理待ちプログラムの実行優先順位を動的に変更し、変更後の優先順位を上記オペレーティングシステムに対し指示することにより、実行中のプログラムおよび待機中のプログラムに対し、優先順位を動的に制御する。

【0034】特許請求の範囲の請求項10に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、ユーザにより定義される終了状態に応じて連続して行なうべき後処置の定義を記録する。停止処理部は、オペレーティングシステムからの終了メッセージにより、プログラムが終了したことを認識し、プログラムの終了した時の終了状態を識別する。また、上記プログラム定義ファイルの中の定義に従い、後処置を実行する。また更に、プログラム停止時に、プログラム名、計算機名、停止日時を含む停止ログをログファイルに記録する。

【0035】特許請求の範囲の請求項11に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、実行中のアプリケーションプログラムごとに使用できる主記憶装置（メモリ）や、補助記憶装置（ディスク）の容量や、演算処理時間の累積値の制限値を記録する。計算機リソース管理部は、上記プログラム定義ファイルに定義された制限値を越えてプログラムが実行されたことを検出し、計算機のリソースが制限値を越えて使用された時に障害と判断し、上記障害回復処理部、および上記障害連絡部に連絡する。

【0036】特許請求の範囲の請求項12に記載の計算機運転管理システムにおいて、プログラム定義ファイルは、ネットワーク上で関連するプログラムの情報を記録する。ネットワーク管理部は、異なる計算機にある運転管理システム間で通信連絡し、上記プログラム定義ファイルからの情報に従い、ネットワークを利用するアプリケーションソフトウェアの動作を管理し、ネットワーク上で関連のあるプログラムの組み合わせを認識する。

【0037】特許請求の範囲の請求項13に記載の計算機運転管理システムにおいて、実行制御部、実行状態監

視部、計算機リソース管理部、停止処理部は一連の主プログラムの中で連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出されて実行される。

【0038】特許請求の範囲の請求項14に記載の計算機運転管理システムにおいて、実行制御部、実行状態監視部、停止処理部は一連の主プログラムの連続処理で実行され、実行優先順位制御部、障害回復処理部、障害連絡部、計算機リソース管理部、ネットワーク管理部については、独立したプログラムとして、主プログラムより必要に応じて呼び出され実行される。

【0039】

【実施例】次に、図面に示す実施例に基づき、本発明について詳細に説明する。

実施例1. 図1は、本発明の計算機システムにおけるソフトウェア運転管理システムを表すものである。図1において、本発明の計算機運転管理システム1は、オペレーティングシステム2とアプリケーションプログラム3の間に介在し、オペレーティングシステム2から見ればアプリケーションプログラム3の1つであり、アプリケーションプログラム3から見ればオペレーティングシステム2の一部と同等に扱えるようなインターフェースを持っており、アプリケーションプログラム3の実行に対し、監視、制御を行う。

【0040】また、上記オペレーティングシステム2は、以下に示した特徴を持つオペレーティングシステムである。

(1) 同時に複数のプログラムを並列に実行できる。

(2) 1つの実行中プログラム（親プログラム）から新しいプログラム（子プログラム）を生成することができる。オペレーティングシステムによる制限を受けずに同時実行プログラム数が増加する。

(3) 同時に実行しているプログラムは、特に指定がなければ、同じ優先順位で実行され、平等に扱われる。実行中の優先順位を動的に変更することができる。

(4) 仮想記憶領域は、有限の大きさを持つが、この限界を越えてプログラムを実行しようとした時、即座にエラー終了となる。

【0041】図2は、図1に示す計算機運転管理システム1の構成図を示す。図2において、計算機運転管理システム11は、上記オペレーティングシステム上で動作し、計算機の主記憶装置（メモリ）上または補助記憶装置（ディスク）上に存在し、実行プログラムの監視、制御を行うプログラムであるプロセスマネージャ12と、補助記憶装置（ディスク）上に存在し、プログラム種別ごとの同時に実行可能なプログラム数の制限値、プログラムの実行優先順位、障害時処理手順、計算機リソースの制限値の情報を記録する記録手段であるプログラム定義ファイル21と、補助記憶装置（ディスク）上に存在

し、プログラムの起動時に、実行の開始を制御されているプログラムの情報を記録する記録手段である実行処理待ちプログラム記録ファイル（要求キュー）22と、および補助記憶装置（ディスク）上に存在し、プログラムの起動、停止、障害時のログを記録するためのログ記録手段であるログファイル23を備える。

【0042】また図2において、プログラム定義ファイル21は、プログラム種別ごとの同時に実行可能なプログラム数と、プログラム実行時の優先順位と、プログラム実行中に障害が発生した時の障害回復手順と、プログラムが正常終了した時の停止処理の手順のデータを含む。上記のデータは全てユーザにより定義される。また、プログラム種別は、ユーザによりあらかじめプログラムのグループ分けがなされたもので、プログラムの種類を表す。該プログラム種別ごとに、同時実行可能プログラム数、優先順位、障害回復手順、停止処理が定義される。上記プログラム定義ファイル中のこれらのデータは、プログラムの起動時、停止時、障害時に参照され、記録されている所定の処理が行われる。

【0043】また図2において、実行処理待ちプログラム記録ファイル（要求キュー）22は、実行を制限された処理待ちのプログラムを待機させるためのバッファエリアで、該実行処理待ちプログラム記録ファイル22中では、プログラム種別ごとに、プログラムの優先順位の高い順に待機プログラムが記録される。すなわち、実行処理待ちプログラム記録ファイル22中では、プログラム種別ごとに複数の処理待ち行列が存在し、該待ち行列の中で優先順位の高い順に待機プログラムが記録される。優先順位の高いほど、先に実行される。

【0044】また図2において、ログファイル23には、プログラムが起動した時に、起動したプログラムのプログラム名、起動日時、計算機名、起動時のプログラム番号を起動ログとして記録される。また、障害発生時の障害ログとして、プログラム名、障害発生日時、計算機名、プログラム番号、障害状況が記録される。障害状況は、オペレーティングシステムからの終了メッセージや、障害発生時の状況を、プロセスマネージャ12により与えられる。また更に、プログラム停止時の停止ログとして、プログラム名、停止日時、計算機名、プログラム番号、終了状態（正常終了か異常終了）が記録される。該記録ログファイルは、オペレータにより参照され、システムの運用状況の把握や障害原因の解明などに利用される。

【0045】また図2において、上記プロセスマネージャ12は、以下の種々の制御部13-20を含む。オペレータまたはプログラムからのアプリケーションプログラム起動の指示に従い、アプリケーションの実行を制御する実行制御部13。

【0046】実行されたアプリケーションプログラムとメッセージのやりとりを行うことにより、該アプリケー

ションプログラムの実行状態を監視し、実行プログラムに障害が発生した時は、所定の障害時処理を行う制御部に連絡する実行状態監視部14。

【0047】CPU、メモリ、ディスク等の計算機リソースの利用状況を、上記プログラム定義ファイル21から読みだした許容値と実際の使用量を比較することにより監視し、障害発生時には、所定の回復手順を行う後述の障害回復処理部と所定の障害連絡を行う障害連絡部に連絡をする計算機リソース管理部15。

【0048】プログラム定義ファイル21から読み出したプログラムの優先順位に従い、実行処理待ちプログラム記録ファイル（要求キュー）22内のプログラムの優先順位と、実行中のプログラムの優先順位を制御する実行優先順位制御部16。

【0049】プログラムが停止した時に、プログラム定義ファイル21に格納されている停止後処理手順に従い停止後処理を行い、停止時ログをログファイル23に記録する停止処理部17。

【0050】障害が発生した時に、プログラム定義ファイル21に格納されている所定の障害回復手順に従い、回復処理を行い、ログファイル23に障害ログを記録する障害回復処理部18。

【0051】障害が発生した時に、プログラム定義ファイル21に格納されている障害回復手順に従い、アラーム発報等の連絡手段により、障害発生後、オペレータに対し迅速に障害の連絡を行う、障害連絡部19。

【0052】また更に、該プロセスマネージャ12は、通信ネットワークを介し、他の計算機上にあるプロセスマネージャ12と連絡し、プログラムの起動、エラー、停止に関するログ情報を交換するネットワーク管理部20を含む。これによって、他の計算機上のプログラムの実行状況が把握でき、1つの計算機によって運転管理を集中的にできることを実現する。

【0053】図3は、本発明における計算機運転管理システム11の各制御部の全体的なつながりを示す機能関連図を表したものである。

【0054】アプリケーションプログラムを実行する時は、オペレーティングシステムではなく、計算機運転管理システム11に実行を依頼する。オペレータまたは他のプログラムから、該計算機運転管理システム11に対しプログラム起動要求が発生すると（S21）、実行制御部13（S22）はプログラム定義ファイル31より実行可能プログラム数を読み出し、実行中のプログラム数と比較し、実行中のプログラム数が、読み込んだ制限値より小さければ、子プログラムとして、オペレーティングシステムに対しプログラムの起動要求を指示する（S23）。

実行中のプログラム数が制限値と同じでさらにプログラム起動要求がある時、実行処理待ちプログラム記録ファイル33にプログラムを待機させる。該実行制御部は、定期的に実行中のプログラム数の監視を行

っており、待機中のプログラムが実行可能になれば、実行処理待ちプログラム記録ファイル33からプログラムをメモリに読み出し、オペレーティングシステムに対しプログラムの起動要求を指示する(S23)。実行制御部13は、プログラムの起動が完了するとログファイル32に起動ログを記録する。

【0055】このように実行されたアプリケーションプログラム3は、計算機運転管理システム1中のプロセスマネージャ12と常に情報交換(通信)ができるように、プロセスマネージャ12と親子の関係を維持した状態で実行される。上記オペレーティングシステム2では、新しくプログラム3が実行された場合、それを実行したプログラム(計算機運転管理システム1中のプロセスマネージャ12)が「親」、新しいプログラム3が「子」と認識される。このように発生した「親子」間の通信手段は、その他の関係のプログラム間通信より大幅に簡単化される特徴がある。プロセスマネージャ12で管理されるアプリケーションプログラムは、常にプロセスマネージャ12に対して「子」の関係を持ち、「子」の「子」すなわち「孫」にはならないように構成することを特徴とする。

【0056】プログラムが起動されると、実行状態監視部14により、実行中のプログラムとの間でメッセージの送受信を行うことにより、実行プログラムの監視を行う(S24)。監視中に障害が発生した時は、障害回復処理部18、障害連絡部19に連絡し、所定の処理を行う(S29、S30)。

【0057】また、プログラム実行中に、オペレータまたは他のプログラムにより実行処理待ちプログラム記録ファイル(要求キュー)33内の待機中のプログラムまたは実行中のプログラムに対し優先順位の変更依頼があった時は、実行優先順位制御部16が実行処理待ちプログラム記録ファイル33内およびメモリ内で実行中のプログラムの優先順位を変更する(S28)。

【0058】実行状態監視部14により、プログラムの実行が監視されるのと同時に、計算機リソース管理部15により、CPU、メモリ、ディスクなどの計算機のリソースの実際の使用量と、プログラム定義ファイル中に格納されたリソースの使用量の制限値とを比較することにより、リソースの監視を行う(S25)。障害が発生した時は、障害回復処理部18および障害連絡部19に通知され所定の処理が行われる(S29、S30)。

【0059】オペレーティングシステムから、プロセスマネージャに対して、プログラムの終了が知らされるまで、実行状態監視(S24)および計算機リソース管理(S25)が行われる。また、ネットワーク管理部20によりネットワークで接続された他の計算機34のプロセスマネージャと間で情報交換を行うことによりネットワークを介して通信をしながら連絡して動作するプログラムの監視も行う(S31)。

【0060】実行状態監視中およびリソース管理において、障害が発生した時は、その情報が障害回復処理部18に与えられ、プログラム定義ファイルに格納されている回復処理手順に従い回復処理が行われるのと同時にログファイル32に障害ログが記録される(S29)、また障害連絡部19によって、障害回復処理手順定義に従い、アラーム等による通報が行われる(S30)。

【0061】オペレーティングシステムから、プロセスマネージャに対して、プログラムの終了が知らされると、停止処理部17によって、プログラム定義ファイルに登録された停止処理が行われ、ログファイルに対して停止ログが書き込まれる(S27)。

【0062】以上、本発明の計算機運転管理システム1において、全体的な各制御部の機能についてのべてきたが、以下に各制御部について、更に詳細な説明を行う。

【0063】図4は、本発明の計算機の運転管理システム11において、プロセスマネージャ12の実行制御部13のフローチャートを示したものである。

【0064】図4において、オペレータまたは、他のアプリケーションプログラムからのプログラム起動要求(S41)は、プロセスマネージャ12に送られる。プロセスマネージャ12は、プログラム定義ファイル21から、該当するプログラムを同時に実行できる数の制限値を読み込む(S42)。同時に実行できるプログラムの制限値としては、(1)そのマシン上で実行できるプログラムの総数、(2)プログラム種別ごとの同時実行数がある。さらに、現在、その計算機(マシン)上で起動されている同一種類のプログラムの数をカウントする。プログラム定義ファイル21から読み込んだ制御値と実行中プログラム数を比較し(S44)、該当するプログラムの数が制限値より少ない場合、そのプログラムは、即時実行可能として実行の手続きを行う(S45)。実行の手続きは、オペレーティングシステム(OS)に対して、当該プログラムをプロセスマネージャの「子」(子プロセス)として起動を指示する。正常に起動が完了すれば、起動したプログラムの名前、日時、計算機の名前、起動時のプログラム番号を起動ログとしてログファイルに23記録する(S46)。

【0065】プログラムの実行時において、現在実行中の当該プログラム数が制限値より多い場合、そのプログラムは、即時実行可能として、実行待ちのバッファエリアである実行処理待ちプログラム記録ファイル(要求キュー)22に記録される(S47)。

【0066】実行処理待ちプログラム記録ファイル(要求キュー)22に待機状態となった実行要求待ちプログラムは、一定時間経過後に定期的に再度実行が可能な状態になったかどうかをプロセスマネージャに12より確認される。現在実行中のプログラム数が制限値より多い状態が続く限り、再実行の条件確認が繰り返される。現在実行中のプログラムが制限値より少なくなるとプログ

ラムの実行手続きがオペレーティングシステムに対して行われる(S45)。

【0067】図5は、本発明の計算機運転管理システムにおいて、実行状態監視部14のフローチャートを示したものである。

【0068】図5において、プロセスマネージャ12は、プログラムの起動要求(S51)に基づき、アプリケーションプログラムをプロセスマネージャ12の「子」として起動する。上記オペレーティングシステムでは、プログラムを起動する場合の起動元を「親」プロセス、起動されたプログラムを「子」プロセスと呼ぶ。これらの「親子」関係にあるプログラム間では、通信手順が簡単になる特徴がある。

【0069】全てのアプリケーションプログラムは、プロセスマネージャ12の「子」としてプログラムが実行され、これらの間には、定期的な通信すなわちメッセージのやりとりを行うことで、アプリケーションプログラムの実行状態をプロセスマネージャ12が集中管理することに特徴がある。

【0070】図5において、実行状態監視部14は、一定時間間隔で、自分の管理下であるアプリケーションプログラムに対して、実行中であることを確認するメッセージを送付する(S52)。メッセージを受け取ったアプリケーションプログラムは、すぐに、どの確認メッセージに対応した応答であるかを示す返答メッセージを実行状態監視部14に返す。プロセスマネージャ12は返信が一定時間内に返ってくることで、アプリケーションプログラムが実行状態であることを確認する(S53)。プロセスマネージャ12はアプリケーションプログラムとの間でこのようなメッセージのやりとりを、オペレーティングシステムからのアプリケーションプログラム終了メッセージを受ける(S54)まで繰り返す行うことで実行状態の監視を行う。

【0071】全てのアプリケーションプログラムは、プロセスマネージャ12からの確認メッセージを受ければ、すぐに返信メッセージを送る仕組みを組込んでいる。この仕組みは、プログラムを作成する時にライブラリとしてサブルーチンを組み込むことで実現する。

【0072】一定時間内に返答メッセージが返ってこない場合(S53)、プロセスマネージャ12は、アプリケーションプログラムに異常が発生したと判断し、障害回復処理部18と障害連絡部19に通知する(S55、S56)。該障害回復処理部18と障害連絡部19は、所定の手順に従い処理を行う。

【0073】このように、実行中のプログラムに異常が生じた時は、計算機運転管理システムがこの異常を検出し、所定の処置を実施する。プログラムで生じる異常には、プログラム自身やオペレーティングシステムが検出できない性格のものが多く、このような計算機運転管理システムが、高信頼性を要求されるシステムの運転時に

において必要不可欠になる。

【0074】ここで、本発明の計算機運転管理システムの実行制御部13、実行状態監視部14において、複数プログラムを同時に実行するための方法を説明する。

【0075】オペレータまたはプログラムより、新規にプログラム起動要求があると、前述したように、プロセスマネージャ12は、子プロセスとしてプログラムを起動する。プログラム起動後は、実行状態監視部14により、このプログラムの実行状態の監視が行われる。この状態になると、プロセスマネージャ12は、新たに、別のプログラムの起動要求を受けつけることが可能になる。さらに別のプログラムの起動要求が発生した時は、同様にして、実行制御部13により、新しいプログラムを起動し、起動後、状態監視を行う。この時点でプロセスマネージャ12は、2つのプログラム実行に関し状態監視を行っており、さらに新しいプログラムの起動要求を受け付け可能状態になる。以下同様にして、同時に実行できるプログラムの数を増やすことができる。

【0076】図6は、本発明の計算機運転管理システムにおいて、実行優先順位制御部16のフローチャートを示したものである。

【0077】図6において、プログラムの起動要求がプロセスマネージャ12に指示される(S61)と、実行優先順位制御部16はプログラム定義ファイル21からグループ別に管理されているプログラムの優先順位と同時実行可能プログラム数を読み込む(S62)。次に実行優先順位制御部16は、現在実行中のプログラムと、同じプログラム種別を持ったプログラムの数を確認する(S63)。現在実行中のプログラム数がプログラム定義ファイルから読み込んだ実行可能数より小さいときは、プロセスマネージャの子プログラムとして、オペレーティングシステムに対し優先順位を付加して起動指示することにより、プログラムを起動する(S65)。プログラムが同時実行可能プログラム数の制限値を越えて新たに実行されようとした時は、即時に実行不可能であるので、プログラムに定義済み優先順位(初期値)を付加し、ディスク上の実行処理待ちプログラム記録ファイル(要求キュー)22に処理待ち状態として一時保管する(S66)。保管されたプログラムは要求キュー22内で他の処理待ちプログラムと優先順位を比較され、優先順位の高い順に要求キュー22内で並べ替えられる(S67)。

【0078】プロセスマネージャ12は、定期的に実行中のプログラムの数をモニタリングし、要求キュー22上で待機中のプログラムが実行可能になった場合は、実行制御部により待機中のプログラムはメモリ上に移動され実行される。

【0079】図7は、また実行優先順位制御部16の、コンソールからのオペレータの指示や、他のプログラムの指示に基づき、実行中のプログラムの優先順位を動的

に変更する機能を示したものである。

【0080】コンソールからのオペレータの指示または、他のプログラムから、実行中のプログラムに対し、優先順位の変更の指示があると（S71）、実行優先順位制御部16は、現在実行中のすべてのプログラムとその優先順位を掌握しており、外部からの新たな優先順位変更依頼に基づき、他のプログラムとの関係を考慮した適切な優先順位を評価する（S72）。

【0081】評価とは、該当プログラムの優先順位が実行中の他のプログラムの実行優先順位と比較して相対的にどの程度かを把握し、優先順位の変更の割合を決定することである。したがって、該当プログラム以外が優先的に処理されている状態で、該当プログラムの優先度を上げる場合は大幅に優先度を上げて、他より優先処理される様にする。メモリ上で実行中のプログラムの優先順位は、そのプログラムを実行するためのCPU（中央演算装置）を使用できる時間（CPU時間）を変化させることにより実現する。

【0082】このような評価に基づき、優先順位は変更され（S73）る。変更後、優先順位は即時有効となり、新しい優先順位に基づいて処理される。この処理はプログラム名と変更された優先順位をオペレーティングシステムに指示することで実現する。

【0083】また、実行優先順位制御部は、実行待機中のプログラムに対しても、コンソールからのオペレータの指示または他のプログラムからの指示により、動的に優先順位を変更できる。実行待機中のプログラムに対し、コンソールからのオペレータの指示または他のプログラムから、優先順位の変更の指示があると、上記と同様に要求キュー内の同じプログラム種別のプログラムの中で優先順位の比較が行われ、優先順位の変更がされる。要求キュー22内では、優先順位の高い順に並べ替えられる。

【0084】図8は、本発明の計算機運転管理システムにおいて、障害回復処理部18のフローチャートを示したものである。

【0085】図8において、障害回復処理部18は、前述の実行状態監視部14により検出されたアプリケーションプログラムの異常状態発生連絡に基づいて起動する（S81）。障害回復処理部18は、プログラム定義ファイルから障害回復処理手順を読み込む（S82）。該障害回復処理手順は、アプリケーションプログラムの種別ごとに定義され、異常時の処置、回復手順、アラーム発報、メール、関連プログラムの処置から構成される。同一プログラム種別のプログラムには同一の回復処理手順が適用される。個々のプログラムに個別の回復処理手順を適用することも可能である。これらのデータは、システム管理者が必要に応じて書き換えることができ、書き換えた時から、プロセスマネージャの再起動なしに新しい手順が有効となり実行可能となる。

【0086】障害回復処理部18は、読み込んだ回復処理手順に従い、回復処理（S84）を忠実に実行する。またこの処理は、他の動作中のアプリケーションプログラムより優先的に実行され、実行時の記録は、ログファイル23にエラーログとして計算機名、プログラム名、処置内容、時刻が残される（S83）。

【0087】図9は、障害回復処理手順の定義の一例を示したものである。図9において、プログラム種別は、プログラムの種類を示す。同じ種類のプログラムでも同時に複数個実行されるが、回復処理は種別で同一になる。

【0088】また図9において、異常時の処置は、実行中のアプリケーションプログラムに異常が生じた場合の処置を示し、処置なし、強制終了、障害回復処理がある。強制終了と障害回復処理は、それぞれ手順を示した回復手順をプログラム定義ファイル21中に定義されている。

【0089】また更に、アラーム発報は、障害発生時にポケットベル、構内アナウンス等でオペレータに自動連絡するかどうかを定義する。メールは、一般にいう電子メールであり、障害発生時に計算機のコンソールにアラームを表示し、関係者の所有する他の計算機に電子メールで通知するかどうかを記述する。これらの障害時における通知機能に関しては、後述する障害連絡機能により実現される。

【0090】また更に、関連プログラムは、該当するプログラムと同時に関連を持ちながら実行しているプログラムグループ、又は、該当プログラムの終了後、連続して起動するプログラムのグループを記述している。該当プログラムグループに属する全てのプログラムに対して、障害回復処理手順がプログラム定義ファイル中に定義しており、これに基づいて実行中の全該当プログラムが処置される。

【0091】図10は、本発明の計算機運転管理システムにおいて、停止処理部17のフローチャートを示したものである。

【0092】図10において、実行中のプログラムを停止した時（S101）、停止したことがオペレーティングシステムからプロセスマネージャの停止処理部17に連絡される（S102）。停止処理部17は、オペレーティングシステムから返されるプログラムの終了ステータスに基づき、正常終了か、異常終了かを判断する（S103）。異常終了であった場合は、障害回復処理部18に連絡され（S107）、それ以降の回復処理を依頼する。正常終了であった場合、停止処理部17は、ログファイル23に、終了したプログラム名、マシン名、日時、プログラム番号、終了ステータスを停止ログとして記録する（S104）。次に、プログラム定義ファイル21から、プログラムに定義された終了後に連続して行うべき処理の手順を読み込み（S105）、その手順を

忠実に実行する(S106)。この後処理を利用して、プログラムの連続処理が実行できる。後処理が終了したことで一連の処理の終了となる。

【0093】図11は、本発明の計算機運転管理システムにおいて、計算機リソース管理部15のフローチャートを示したものである。

【0094】図11において、計算機リソース管理部15は、管理下である実行中の全プログラムに対して、プログラムが使用しているCPUタイムの累積値をチェックする(S111)。実際のCPUの累積値とプログラム定義ファイルから読み込んだ制限値(所定値)とを比較する(S112)。比較した結果、制限値を超えていた時は、障害処理部18、障害連絡部19に連絡する(S121、S122)。

【0095】CPUタイムの確認で異常がなかった時は、次にメモリの使用量のチェックを行う(S114)。実際のメモリの使用量とプログラム定義ファイル21から読み込んだ制限値(所定値)とを比較する(S115)。比較した結果、制限値を超えていた時は、障害回復処理部18、障害連絡部19に連絡する(S121、S122)。

【0096】同様に、ディスクの使用量のチェックを行う(S117)。実際のディスクの使用量とプログラム定義ファイル21から読み込んだ制限値(所定値)とを比較する(S118)。比較した結果、制限値を超えていた時は、障害回復処理部18、障害連絡部19に連絡する(S121、S122)。

【0097】リソースの制限値は、ユーザにより、プログラム定義ファイル21の中に、各プログラム種別ごとに、予め使用できるリソースの量、すなわち、CPUの累積時間、メモリ容量、ディスク容量が定義されている。リソースの制限値をオーバーした場合の処置方法もユーザにより、プログラムごとに該プログラム定義ファイル21に定義されている。

【0098】以下に、本発明の計算機運転管理システムにおける、ネットワーク管理部20について説明する。

【0099】計算機どうしが、互いにネットワークで接続され、また管理下のアプリケーションプログラムが、互いに通信しながら、連携して動作するアプリケーションを持つ計算機において、該計算機上に存在するプロセスマネージャ12どうしも互いに通信することで情報交換し、複数の計算機上で動作する大規模アプリケーションプログラムシステムを運転管理するのが本機能の目的である。

【0100】異なる計算機上のプロセスマネージャ12間において交換される情報は、プログラムの起動、停止、エラーに関する情報を記録したログファイル情報、計算機間で通信しながら動作しているアプリケーションプログラムの動作状況、プログラム定義ファイル(優先順位、障害回復手順等)に関する情報である。ログファ

イルには、それが発生した計算機名が記録されている。これにより、どの計算機で発生したログかを判断することができ、1台の計算機で集中的に監視することができる。計算機間で通信しながら動作しているアプリケーションプログラムの動作状況も、それぞれの計算機上のプロセスマネージャで管理され、プロセスマネージャ間で通信連絡が定期的に行われることで、プログラム全体が矛盾なく管理できる。プログラムの優先順位、障害回復手順などのプログラム定義ファイルの情報は、必要に応じて1台の計算機上に入力された情報が複数台の上に複写されることで、一元管理できる。

【0101】図12は、本発明の計算機運転管理システムにおける、ネットワーク管理部20のフローチャートを示したものである。図12において、計算機で実行中のアプリケーションプログラムに対して、プロセスマネージャ12がアプリケーションプログラムの状態監視のため、アプリケーションプログラムに対して、メッセージを送信する(S121)。プロセスマネージャ12が、一定時間内にアプリケーションプログラムからの返信メッセージを受信できれば(S122)、メッセージのやりとりによる状態監視を定期的に行う。

【0102】プロセスマネージャ12が、一定時間内にアプリケーションプログラムからの返信メッセージを受信できなかった時は(S122)、通信相手のプログラムの実行状態を確認する。プロセスマネージャ12はあらかじめ、プログラム定義ファイル21からの関連プログラムの定義により、実行プログラムの通信相手先の計算機とプログラムがどれであるかを認識している。通信相手先のプロセスマネージャとプログラムの間で状態監視のためのメッセージのやりとりを行い(S123)、該プロセスマネージャが該プログラムからの返信メッセージを正常に受信できれば(S124)、プロセスマネージャは、障害がネットワーク部で生じたと判断する(S127)。該プロセスマネージャが該プログラムからの返信メッセージを受信できなければ、該プロセスマネージャは、障害が管理下のプログラムに発生したと判断し(S125)、エラーログとして、ログファイル23に書き出す(S126)。

【0103】図13、図14は、障害連絡部19のフローチャートである。障害連絡部19は、障害発生時に所定の手順に従い障害をオペレータ等に連絡する障害連絡機能と、ネットワークを介して他の計算機で発生した障害の発生を検知する障害監視機能を有する。

【0104】図13は障害連絡機能について説明したものである。障害連絡部19は、実行状態監視部14、計算機リソース管理部15より、障害の通知を受ける(S131)とプログラム定義ファイル21中からの障害回復手順を読み込み(S132)、該障害回復処理手順に従い、障害をオペレータに連絡すべく、そのメッセージ内容を汎用アラーム、電子メール、集中監視盤等に出力

する(S133)。

【0105】上記汎用アラームとは、構内放送システムで、電子音によりアナウンスが行われる。上記電子メールは、計算機の電子メールシステムのことで、オペレータ等の関係者の端末に障害発生との連絡のための電子メールを送信する。集中監視盤とは、計算機システムコンソールまたは、モニタ画面であり、一箇所に集中してメッセージが出力できる。同時にディスク上のファイル、プリンタにも記録が出力できる。

【0106】図14において、障害連絡部19の障害監視機能により障害検出後、障害連絡機能により通報を行う動作を示したものである。障害監視機能は、ネットワーク上の計算機におけるログファイル23の内容を定期的に監視し、新規にエラーログが発生したかどうかを確認する(S141)。新規のエラーログの発生が確認できなければ(S142)、定期的に、新規エラーログの発生の有無を確認する(S141)。新規にエラーログの発生が確認できれば(S142)、発生ログの内容を確認(S143)する。障害監視機能により、エラーログの発生が確認されると、上記障害連絡機能により、プログラム定義ファイル21から障害回復手順を読み込み(S144)、プログラム定義ファイル21に記録された所定の手順に従い、アラームの通報を行う(S145)。

【0107】すなわち、障害連絡部19では、ネットワークで接続された計算機上にあるログファイル23を、ネットワークで接続されたプロセスマネージャ12を介して、集中的に管理する機能であり、障害を通報する障害連絡機能とネットワーク上の障害を監視する障害監視機能を含む。エラーログに関しては、ログが発生したことをトリガーとして、発生時に障害連絡機能が起動する。障害監視機能は、予めプログラム定義ファイルで定義された範囲の計算機上で発生する、定義された種類のエラーログを監視する。

【0108】実施例2. 図15は、本発明の計算機運転管理システムにおける第2実施例の各制御部の全体的なつながりを示す機能関連図を表したものである。

【0109】図3において、本発明の第1実施例では、計算機リソース管理機能を主プログラムの連続処理として示したが、図15においては、計算機リソース機能を独立したプログラムとして、主プログラムの連続処理から外し、必要に応じて実行するように構成されている。

【0110】また図15において、各制御部の制御機能については、実施例1で記載した内容と同様のものである。図15において、本発明の計算機運転管理システム1において、アプリケーションプログラム3を実行する時は、オペレーティングシステム2ではなく、計算機運転管理システム1に実行を依頼する。オペレータまたは他のプログラムから、該計算機運転管理システム1に対しプログラム起動要求が発生すると(S201)、実行

制御部13(S202)はプログラム定義ファイル201より実行可能プログラム数を読み出し、実行中のプログラム数と比較し、実行中のプログラム数が、読み込んだ制限値より小さければ、子プログラムとして、オペレーティングシステム2に対しプログラムの起動要求を指示する(S203)。実行中のプログラム数が制限値と同じか大きければ、実行処理待ちプログラム記録ファイル203にプログラムを待機させる。該実行制御部13は、定期的に実行中のプログラム数の監視を行っており、待機中のプログラムが実行可能になれば、実行処理待ちプログラム記録ファイル203からプログラムをメモリに読み出し、オペレーティングシステム2に対しプログラムの起動要求を指示する(S203)。実行制御部13は、プログラムの起動が完了するとログファイル202に起動ログを記録する。

【0111】プログラムが起動されると、実行状態監視部14により、実行中のプログラムとの間でメッセージの送受信を行うことにより、実行プログラムの監視を行う(S204)。監視中に障害が発生した時は、障害回復処理部18、障害連絡部19に連絡し、所定の処理を行う(S209、S210)。

【0112】また、プログラム実行中に、オペレータまたは他のプログラムにより実行処理待ちプログラム記録ファイル(要求キュー)203内の待機中のプログラムまたは実行中のプログラムに対し優先順位の変更依頼があった時は、実行優先順位制御部16が実行処理待ちプログラム記録ファイル203内およびメモリ内で実行中のプログラムの優先順位を変更する(S207)。

【0113】実行状態監視部14は必要に応じ、計算機リソース管理部15を呼び出し、該計算機リソース管理部15により、CPU、メモリ、ディスクなどの計算機のリソースの実際の使用量と、プログラム定義ファイル21中に格納されたリソースの使用量の制限値とを比較することにより、リソースの監視を行う(S208)。障害が発生した時は、障害回復処理部18および障害連絡部19に通知され所定の処理が行われる(S209、S210)。

【0114】オペレーティングシステム2から、プロセスマネージャ12に対して、プログラムの終了が知らされるまで、実行状態監視(S204)が行われる。また、ネットワーク管理部20によりネットワークで接続された他の計算機204のプロセスマネージャ12と間で情報交換を行うことによりネットワークを介して通信をしながら連絡して動作するプログラムの監視も行う(S211)。

【0115】実行状態監視中およびリソース管理において、障害が発生した時は、その情報が障害回復処理部18に与えられ、プログラム定義ファイルに格納されている回復処理手順に従い回復処理が行われるのと同時にログファイル202に障害ログが記録される(S20

9)、また障害連絡部19によって、障害回復処理手順定義に従い、アラーム等による通報が行われる(S210)。

【0116】オペレーティングシステム2から、プロセスマネージャ12に対して、プログラムの終了が知らされると、停止処理部17によって、プログラム定義ファイル21に登録された停止処理が行われ、ログファイル202に対して停止ログが書き込まれる(S206)。

【0117】

【発明の効果】本発明は、同時に実行できるプログラム数をオペレーティングシステムに対し、制限、監視することにより、計算機の主記憶装置(メモリ)、補助記憶装置(ディスク)、演算処理装置(CPU)を効果的に使用することで、以下の効果がある。

【0118】(1)同時に実行できるプログラム数をオペレーティングシステムに対し制限し、メモリやディスクの不足が生じないようにプログラムの実行を管理するため、計算機上で同時に複数のプログラムが実行された場合に、メモリ不足による処理速度の異常低下を防ぎ、パフォーマンスの向上が期待できる。

【0119】(2)計算機上で同時に複数のプログラムが実行された場合に、メモリやディスクの容量が一時的な不足による、一部のプログラムの強制的な異常終了や、システムの機能の一部の欠落を防ぎ、プログラムの異常終了のないシステムの高信頼化が実現できる。

【0120】(3)実行中のアプリケーションプログラムに不具合があり、異常終了した場合も、一括管理したエラーログを参照することで原因となったプログラムとタイミングを追求できる。さらにネットワークを利用するプログラムや、複数のプログラムが協調しながら動作するシステムでも不具合の追求ができる。

【0121】(4)動作中に処理の異常が生じたにもかかわらず、エラーが発行しないような不完全なプログラムでも、異常が生じたことを検出できる。

【0122】(5)実行するときのプログラムの優先処理順序を定めることができ、各プログラムに必要なリアルタイム性に応じた処理ができる。プログラムの重要性に応じた重み付けができ、重要なプログラムを優先的に処理できる。

【0123】(6)プログラムが異常終了した場合、それに対する回復処理を定義し、自動実行することができるため、オペレータのトラブル対策処置を大幅に改善できる。

【図面の簡単な説明】

【図1】本発明の計算機運転管理システムのオペレーティングシステムとアプリケーションプログラムとの関係を示した図。

【図2】本発明の計算機運転管理システムにおける構成図。

【図3】本発明の計算機運転管理システムにおける第一実施例の各制御部の機能構成図。

【図4】本発明の計算機運転管理システムにおける実行制御部のフローチャート。

【図5】本発明の計算機運転管理システムにおける実行状態監視部のフローチャート。

【図6】本発明の計算機運転管理システムにおいて、プログラム起動要求時における実行優先順位制御部のフローチャート。

【図7】本発明の計算機運転管理システムにおける、実行中プログラムの優先順位を動的に変更する時の実行優先順位制御部のフローチャート。

【図8】本発明の計算機運転管理システムにおける障害回復処理部のフローチャート。

【図9】障害回復処理手順定義の例。

【図10】本発明の計算機運転管理システムにおける停止処理部のフローチャート。

【図11】本発明の計算機運転管理システムにおける計算機リソース管理部のフローチャート。

【図12】本発明の計算機運転管理システムにおけるネットワーク管理部のフローチャート。

【図13】本発明の計算機運転管理システムにおける、障害連絡機能を表す障害連絡部のフローチャート。

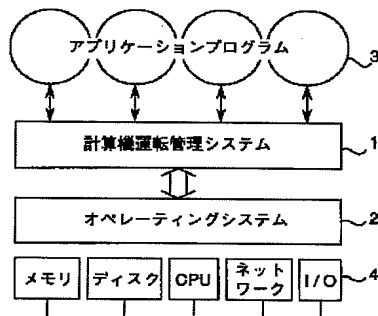
【図14】本発明の計算機運転管理システムにおける、障害連絡機能と障害監視機能を表す障害連絡部のフローチャート。

【図15】本発明の計算機運転管理システムにおける、第二実施例の各制御部の機能構成図

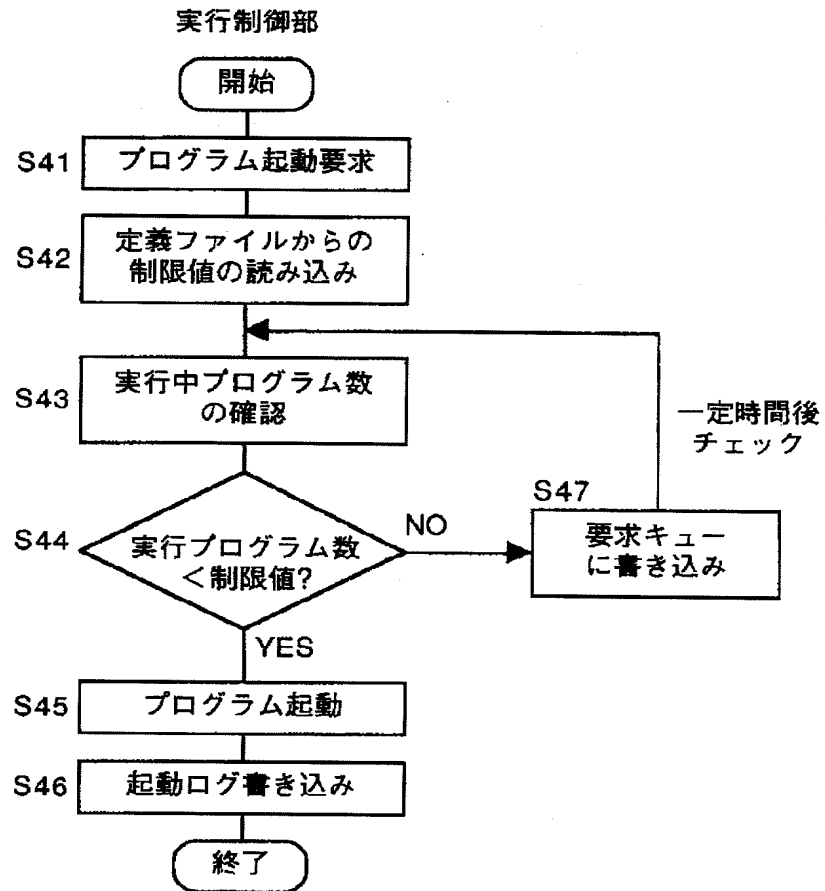
【符号の説明】

1, 11 計算機運転管理システム、2 オペレーティングシステム、3 アプリケーションプログラム、4 計算機ハードウェア、12 プロセスマネージャ、13 実行制御部、14 実行状態監視部、15 計算機リソース管理部、16 実行優先順位制御部、17 停止処理部、18 障害回復処理部、19 障害連絡部、20 ネットワーク管理部、21, 31, 201 プログラム定義ファイル、22, 33, 203 実行処理待ちプログラム記録ファイル、23, 32, 202 ログファイル。

【図1】



【図4】

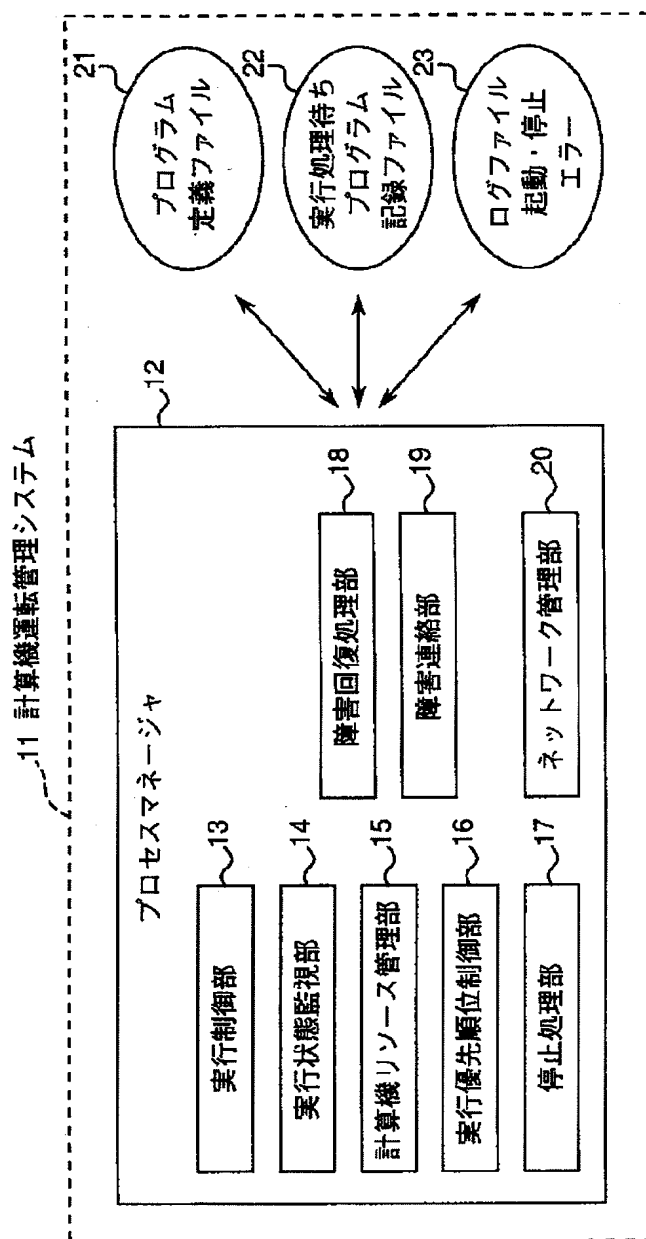


【図9】

障害回復処理手順定義

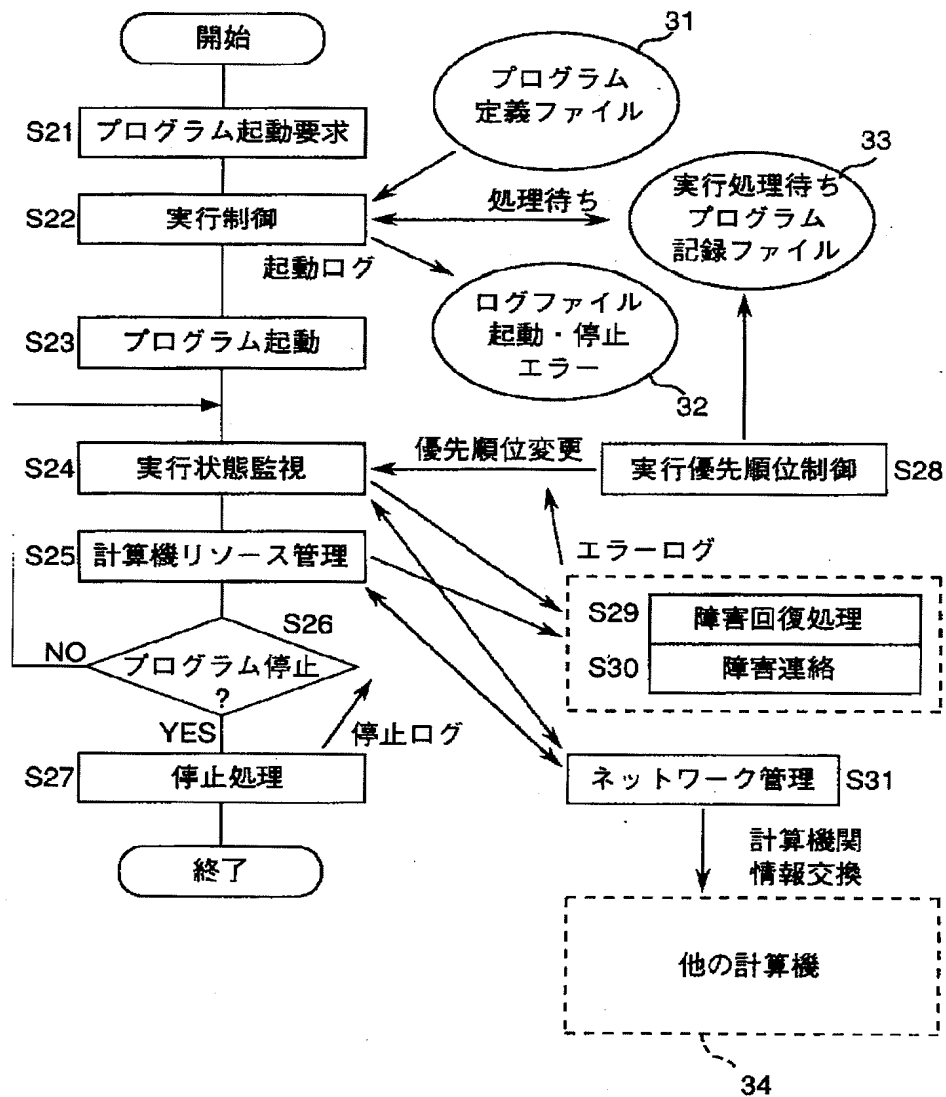
プログラム種別	異常時の処置	回復手順	アラーム発報	メール	関連プログラム	関連プログラムの処置
アプリ1	なし	なし	あり	なし	グループ1	なし
アプリ2	障害回復処理	手順1	あり	送付	なし	関連処理手順4
アプリ3	強制終了	手順7	なし	送付	グループ2	関連処理手順6

【図2】

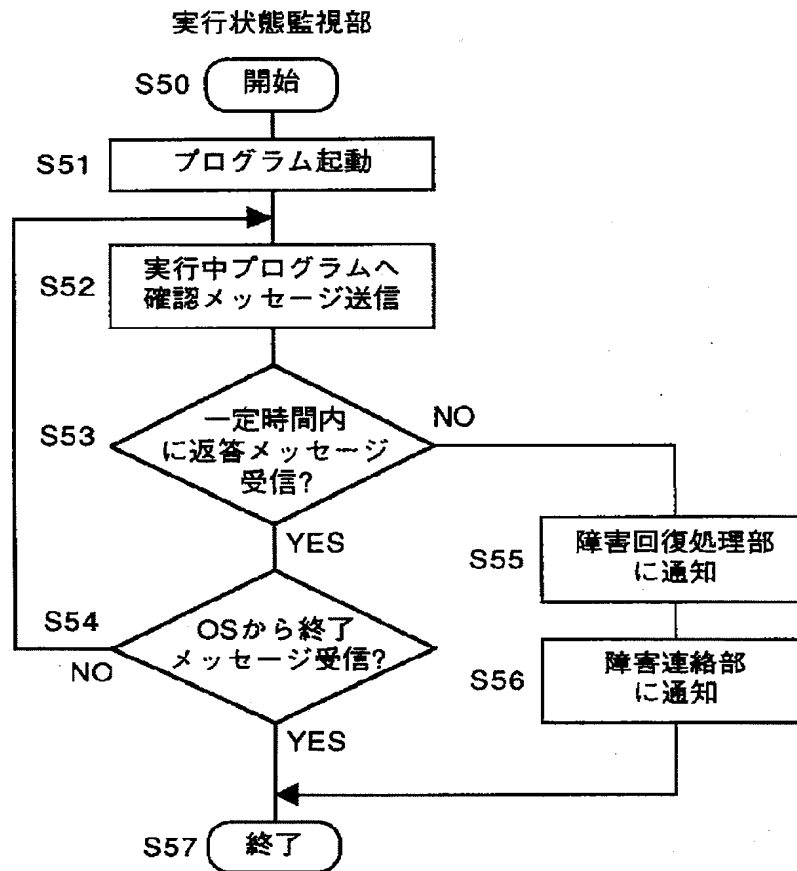


【図3】

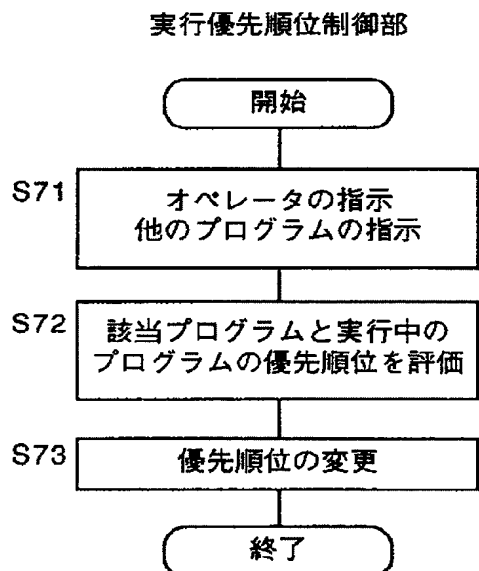
計算機運転管理システム



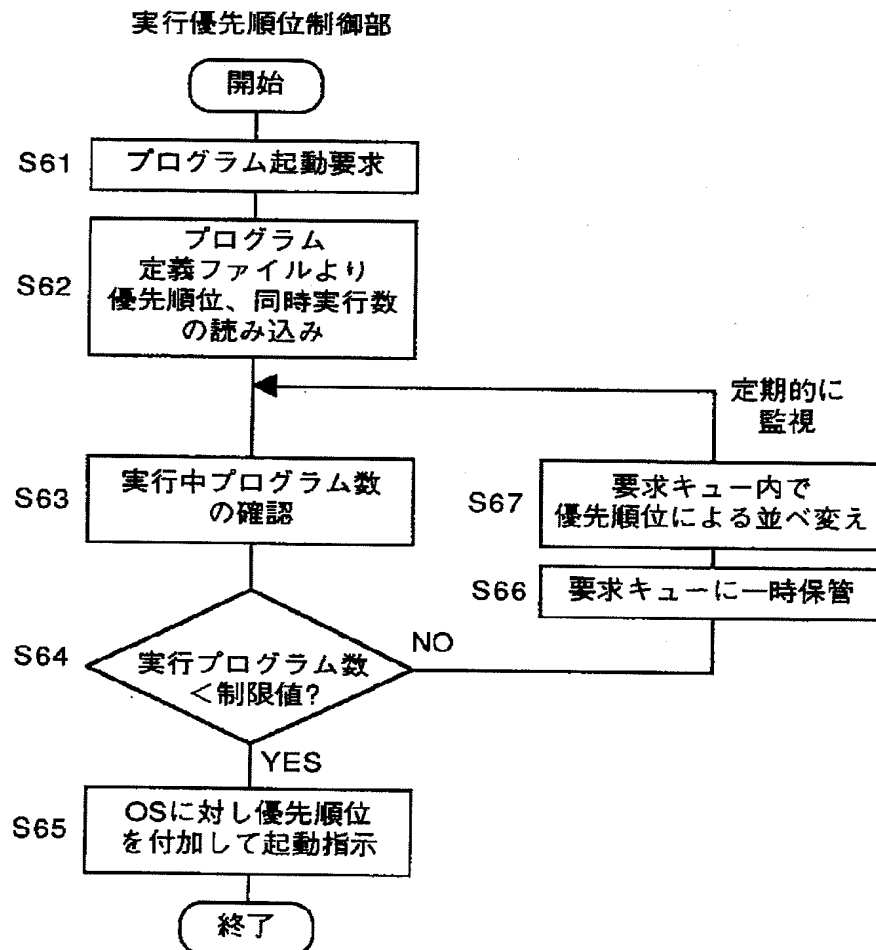
【図5】



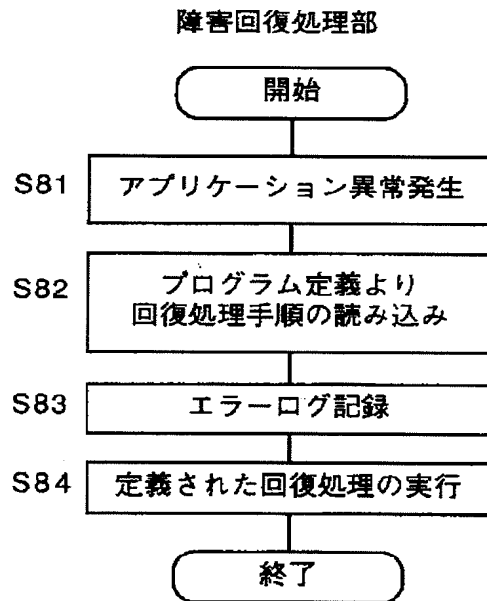
【図7】



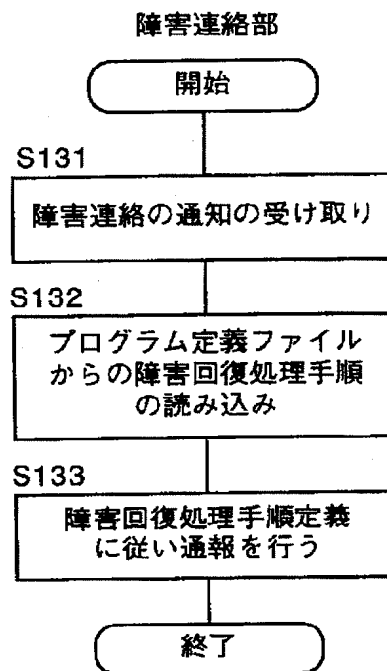
【図6】



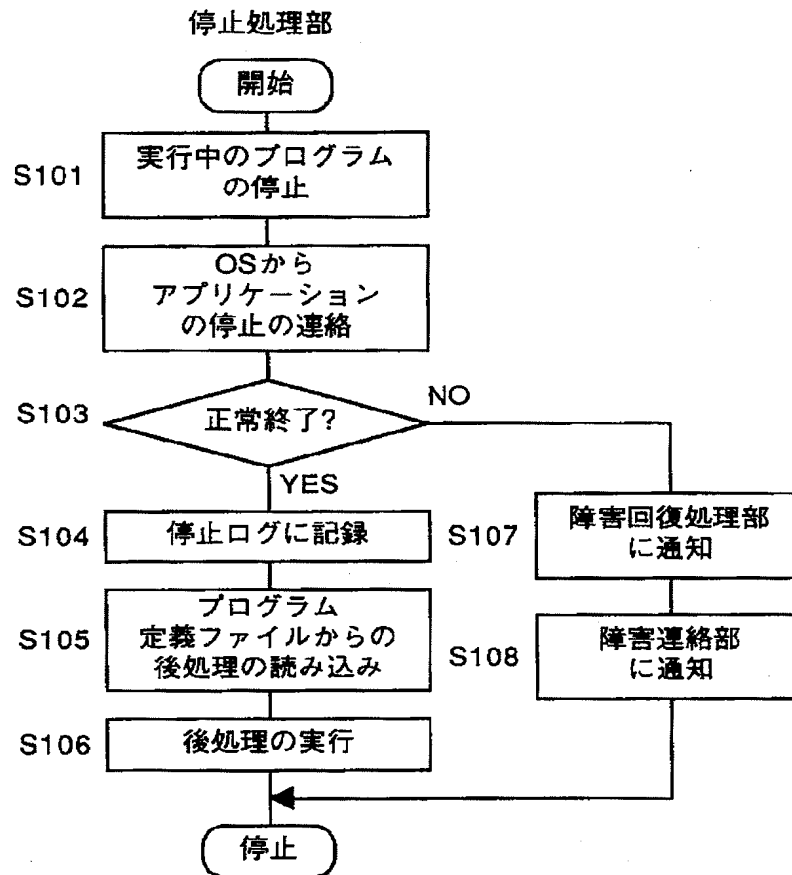
【図8】



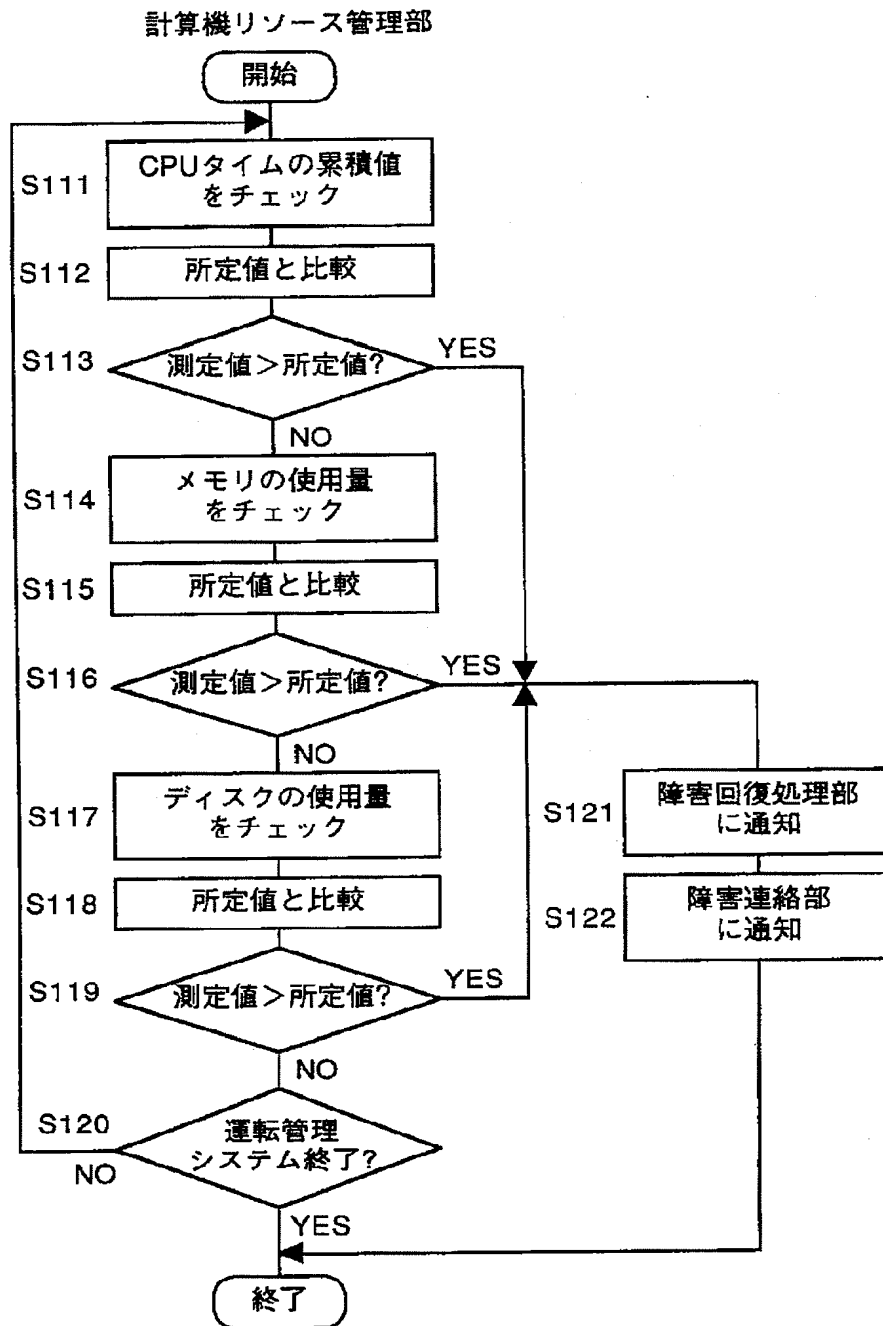
【図13】



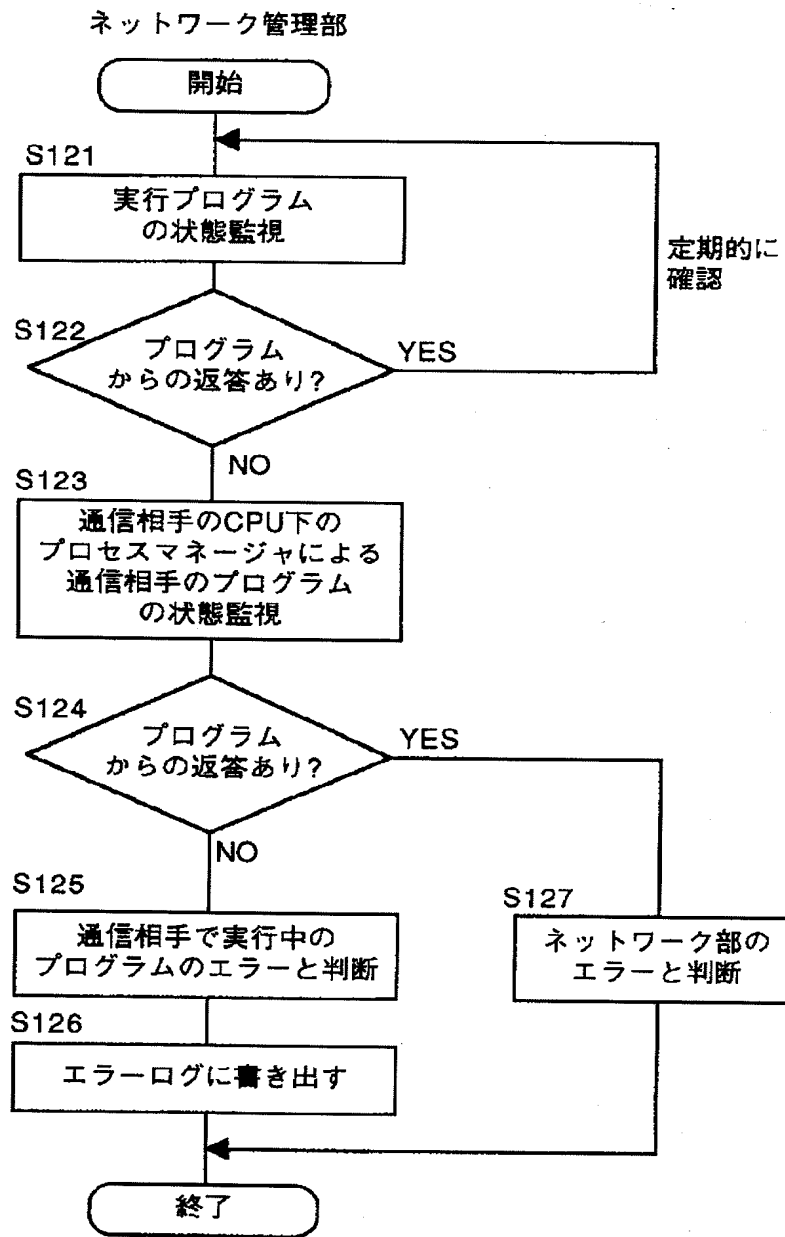
【図10】



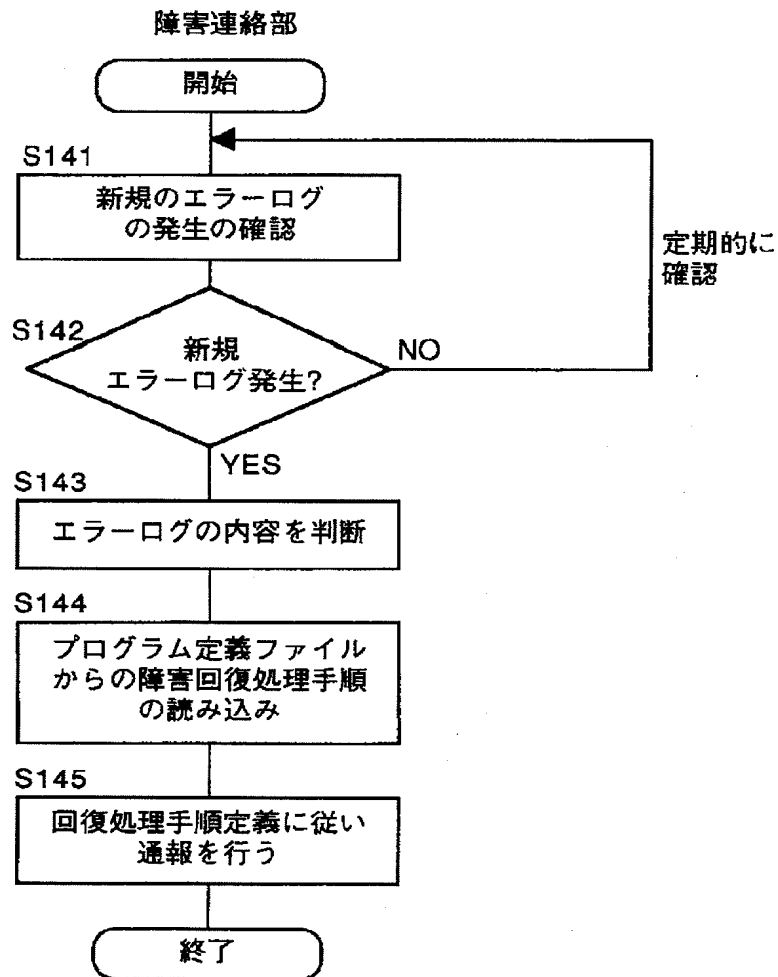
【図11】



【図12】



【図14】



【図15】

計算機運転管理システム

